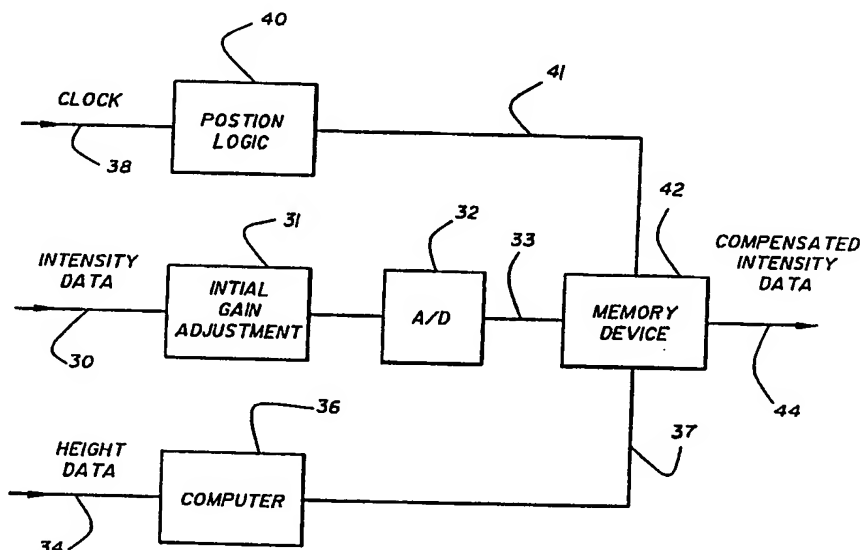


INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification ⁵ : G06K 9/38	A1	(11) International Publication Number: WO 93/04442 (43) International Publication Date: 4 March 1993 (04.03.93)
(21) International Application Number: PCT/US92/07054 (22) International Filing Date: 24 August 1992 (24.08.92) (30) Priority data: 07/749,190 23 August 1991 (23.08.91) US (71) Applicant: UNITED PARCEL SERVICE OF AMERICA, INC. [US/US]; 400 Perimeter Center, Terraces North, Atlanta, GA 30346 (US). (72) Inventor: SHAH, Govind ; 11 Amherst Way, Princeton Junction, NJ 08550 (US). (74) Agents: DROBILE, James, A. et al.; 1600 Market Street, Suite 3600, Philadelphia, PA 19103 (US).		(81) Designated States: AT, AU, BB, BG, BR, CA, CH, CS, DE, DK, ES, FI, GB, HU, JP, KP, KR, LK, LU, MG, MN, MW, NL, NO, PL, RO, RU, SD, SE, European patent (AT, BE, CH, DE, DK, ES, FR, GB, GR, IE, IT, LU, MC, NL, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, ML, MR, SN, TD, TG). Published <i>With international search report.</i>

(54) Title: METHOD AND APPARATUS FOR COMPENSATION FOR NON-UNIFORM ILLUMINATION



(57) Abstract

A method for illumination compensation in a system for processing of signals representing reflected light intensities from a surface includes the steps of receiving reflected light intensity information in the form of more than one intensity value (33) and receiving, associated with each one of the reflected light intensity values, a value of at least one variable (37, 41). The method further includes remapping (42) each of the reflected light intensity values to one or more than one compensated reflected light intensity values, including remapping any of the reflected light intensity values equal to or less than a preselected threshold minimum reflected light intensity value associated with the value of each of the variables for that reflected light intensity value, to a minimum level output, and remapping any of the reflected light intensity values equal to or greater than a preselected threshold maximum reflected light intensity value, associated with the value of each of the variables for that reflected light intensity value, to a maximum level output.

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AT	Austria	FI	Finland	MN	Mongolia
AU	Australia	FR	France	MR	Mauritania
BB	Barbados	GA	Gabon	MW	Malawi
BE	Belgium	GB	United Kingdom	NL	Netherlands
BF	Burkina Faso	GN	Guinea	NO	Norway
BG	Bulgaria	GR	Greece	NZ	New Zealand
BJ	Benin	HU	Hungary	PL	Poland
BR	Brazil	IE	Ireland	PT	Portugal
CA	Canada	IT	Italy	RO	Romania
CF	Central African Republic	JP	Japan	RU	Russian Federation
CG	Congo	KP	Democratic People's Republic of Korea	SD	Sudan
CH	Switzerland	KR	Republic of Korea	SE	Sweden
CI	Côte d'Ivoire	LI	Liechtenstein	SK	Slovak Republic
CM	Cameroon	LK	Sri Lanka	SN	Senegal
CS	Czechoslovakia	LU	Luxembourg	SU	Soviet Union
CZ	Czech Republic	MC	Monaco	TD	Chad
DE	Germany	MG	Madagascar	TG	Togo
DK	Denmark	ML	Mali	UA	Ukraine
ES	Spain			US	United States of America

METHOD AND APPARATUS FOR COMPENSATION FOR
NON-UNIFORM ILLUMINATION

Background of the Invention.

This invention relates to compensation for non-
5 uniformity in illumination of a surface in a system that
takes information in optical form from the surface and
converts such information to electronic form and in par-
ticular to compensating for variations in illumination in
systems for reading labels on articles, such as packages
10 or letters, being moved at high speed along a conveyor
belt past a video camera.

In the shipping of parcels and packages, it is
necessary, at various points in the process, to obtain
various information that may be encoded on a label. Such
15 information includes address information, shipper informa-
tion, and the like. Encoded labels also have application
in other fields, such as manufacturing. Uses for informa-
tion encoded on a label may include, by way of example,
sorting packages further along the conveyor belt, deter-
20 mining the quantity of items going to a particular loca-
tion, and recording the movement of packages so as to
provide updates in response to customer inquiries. The
rapid obtaining of information from a label may be facili-
tated by an automated system for converting optically
25 encoded information on the labels to electrically encoded
information.

In modern terminals for sorting of packages,
packages are loaded on a conveyor belt moving at speeds of
up to 100 inches per second and moved to various points
30 for sorting. Such conveyor belts are ordinarily three to
four feet wide. A common width is 42 inches. Packages
may be loaded onto such conveyor belts anywhere across
their widths, and in any orientation. Packages of differ-
ing sizes may be shipped along a single conveyor belt. As
35 a result when a package passes through a zone for reading
of label information, the label may be at any point across
the width of the conveyor belt and at any height within a
range of heights above the surface of the conveyor belt.

Reflected light intensity is commonly converted to electronic form using a charge coupled device ("CCD"). An appropriate system of lenses focuses reflected light onto a CCD array. The CCD generates a signal, representing an intensity value, on a pixel-by-pixel basis, for the zone for reading label information. Each pixel represents an area viewed by the CCD. Signal processing equipment then converts the intensity value for each pixel to a signal representing one of the possible optical properties of the label. Such optical properties may be white and black, as well as various shades of gray. The optical property is established by comparing the intensity value to one or more reference threshold values.

It is difficult to provide an illumination system that can uniformly illuminate the entire zone through which labels may pass. As noted above, this zone is three to four feet wide and three to four feet high. In fact, the intensity of illumination may vary across the width of the zone at any given height. The intensity of illumination will ordinarily also vary with height. The pattern of intensity across the width of the zone is not the same at different heights. As a result, a CCD or other apparatus for converting reflected light intensity values into electrical signals may produce signals representing different intensity values for the same optical property depending on the location of the label in either height or position across a conveyor belt. As a result of such variations in intensity values, actual threshold values for distinguishing different optical properties from one another will vary depending on the location of the label. If reference threshold values cannot be adjusted to compensate for this effect, errors in reading labels will occur. For example, a white area of a label in a region of low illumination may give a intensity signal that is below a fixed threshold for black or gray. This will result in errors in reading of labels. Such errors of course cause misreading of information on a label.

Objects of the Invention.

Accordingly, it is an object of this invention to provide a method and an apparatus for correcting signals representative of information stored on a surface in optical form detected under conditions of non-uniformity of illumination of the surface.

It is a further object of this invention to provide such a method and apparatus for use in a system in which labels are placed on packages of varying heights.

It is a further object of this invention to provide a method and apparatus for correcting signals representing label information in which labels are placed on packages that are moving on a conveyor belt.

It is a particular object of this invention to provide such a method and apparatus for use where the information is stored on the surface in white areas and black areas.

Additional objects and advantages of the invention will become evident from the detailed description of a preferred embodiment below.

Summary of the Invention.

A method for illumination compensation for use in a system for reading surfaces having at least black and white areas, includes the steps of storing reference reflected light intensity information corresponding to at least black and white areas in the form of the plurality of intensity values, each one of the intensity values having an address defined by a value of at least one variable, calculating from the reference reflected light intensity information more than one black threshold value and more than one white threshold value each of which threshold values is associated with a selected value of each of the variables. The method further includes the steps of receiving operational reflected light intensity information in the form of more than one operational reflected light intensity value, each of the operational reflected light intensity values having an address defined by a value of each of the variables, remapping each of the operational reflected light intensity values to one of at

least two output compensated reflected light intensity values, including the steps of remapping any of the operational reflected light intensity values equal to or less than the black threshold value for its address to a black level output, and remapping any of the operational reflected light intensity values equal to or greater than the white threshold value for its address to a white level output, and producing an output signal including the output compensated reflected light intensity values.

10 A method for illumination compensation in a system for processing of signals representing reflected light intensities from a surface includes the steps of receiving reflected light intensity information in the form of at least two intensity values, receiving, associated with each of the reflected light intensity values, a value of at least one variable, and remapping each of the reflected light intensity values to one of at least two compensated reflected light intensity values, including the steps of remapping any of the reflected light intensity values equal to or less than a preselected threshold minimum reflected light intensity value associated with the value of each of the variables for the reflected light intensity value to a minimum level output and remapping any of the reflected light intensity values equal to or greater than a preselected threshold maximum reflected light intensity value associated with the value of each of the variables for the reflected light intensity value to a maximum level output.

30 An apparatus for illumination compensation in a system for processing of signals representing reflected light intensities from a surface includes means for receiving reflected light intensity information in the form of more than one intensity value, means for receiving, associated with each one of the reflected light intensity values, a value of at least one variable, and memory means for remapping each of the reflected light intensity values to one of at least two compensated reflected light intensity values, including memory means for remapping any of the reflected light intensity values equal to or less than

a preselected threshold minimum reflected light intensity value associated with the value of each of the variables for the reflected light intensity value to a minimum level output and memory means for remapping any of the reflected light intensity values equal to or greater than a preselected threshold maximum reflected light intensity value associated with the value of each of the variables for the reflected light intensity value to a maximum level output.

Brief Description of the Drawings.

Figure 1 is a simplified perspective view of a system for use with a method and apparatus according to the invention.

Figure 2 is a block diagram of a circuit according to the invention.

Figure 3 is a partial view of an calibration bar for use in connection with the invention.

Figure 4 is a flow chart of the overall process of calibration in connection with a method and apparatus according to the invention.

Figure 5 is a graph showing reference reflected light intensity data versus horizontal position before simplification.

Figure 6 is a flow chart showing the steps of reflected light intensity data simplification.

Figure 7 is a graph showing reference reflected light intensity data versus horizontal position after data simplification.

Figure 8 is a graph showing output reflected light intensity values versus input reflected light intensity values as remapped according to the invention.

Detailed Description of a Preferred Embodiment.

With reference to Figure 1, there is shown an example of an apparatus for reading surfaces having information, such as reading label information on articles being moved on a conveyor belt. Such an apparatus may be used in connection with an illumination compensation apparatus of the invention. There is shown a section of a conveyor belt 60. Conveyor belt 60 is moving in the direction of arrow 61. Box 62 is on the conveyor belt 60.

Box 62 has height h. Box 62 has label 64 on a top surface thereof. For clarity, support members are not shown. It will be clear to those of skill in the art of design of mechanical supports that mechanical supports can easily be provided.

In moving on conveyor 60 in the direction shown in the arrow, an article first encounters height sensing means 66. Height sensing means 66 includes two vertical apparatuses 68 and 70 disposed opposite each other across the belt. Apparatuses 68 and 70 may be any one of a number of commercially-available light sensors. One of apparatuses 68 is an emitter, and the other is a detector. Height sensing apparatus 66 is used to ascertain the height h of package 62.

A package 62 next moves through illumination zone 72. Illumination zone 72 is a generally rectangular, planar region of high-intensity illumination provided across the width of the belt up to a selected height above the surface of the belt. The illumination is provided by illumination apparatus 74, which is located over the illumination zone. Light reflected from a box 62 proceeds upward to mirror 76 and then is reflected to light-detecting device 78. Vertical uprights are provided on each side of the illumination zone 72. Each vertical upright has a number of equally-spaced horizontally-protruding pegs 82 extending therefrom. Each peg 82 corresponds to a peg 82 at the same height on the opposite one of the vertical uprights 80. The pegs 82 support the calibration bar. The use of the calibration bar is explained below in connection with the discussion of calibration.

The circuitry of the invention will now be explained with reference to Figure 2. Two types of information are received by the apparatus of the invention. A first type of information is reflected light intensity information from a light-detecting device. In a preferred embodiment, the light-detecting device is a CCD. This reflected light intensity information is transmitted along line 30 to initial gain unit 31. Reflected light intensity data is transmitted by the CCD as signals corresponding

to discrete pixels. The reflected light intensity information is in the form of a series of discrete values or reflected light intensity values. Each discrete value corresponds to a particular position along a scan line.

5 A second type of information is information relating to the values of the variables, or variable value information, upon which the illumination depends. In the disclosed embodiment of the invention, there are two variables upon which the illumination depends. One such
10 variable is height. Height information is transmitted along line 34 to a computer 36.

 The second such variable is horizontal position. The horizontal position of any item of reflected light intensity is preferably derived from a clock signal associated with the reflected light intensity information
15 transmitted by the light-detecting device. A clock signal from the light-detecting device is transmitted along line 38 to position logic unit 40. The light-detecting device also transmits a signal to the position logic unit 40
20 representing the beginning of a new scan line. From the clock signal and the new scan line signal, position logic unit 40 provides a position signal. The unit may be an Altera 5032. The providing of such a signal is conventional in the art. The position signal is sent simulta-
25 neously with each intensity value. The position signal counts from zero on each scan line across the illumination zone. The position signal value is changed a certain number of times in each scan line depending on detail of compensation in the apparatus. This is explained below in
30 more detail where the look-up tables are discussed.

 Computer 36 is shown in Figure 2. As noted above, an input of computer 36 receives height information. Computer 36 has the function of processing height information in the system of the invention. One output of
35 computer 36 is coupled to an input of initial gain unit 31. This output is a gain signal. In a preferred embodiment, this gain signal is an 8-bit digital signal. This gain signal is used as an adjustment for gain variation relating only to height. It has been found that reflected

light intensity varies greatly with height. It is believed that this variation with height results from two causes. First, as the distance from the illumination means increases, the intensity of illumination falls off rapidly. Second, as the distance from the focusing means of the light detecting device increases, a given area on a surface corresponds to a smaller area on the light-detecting device. Accordingly, independent of intensity of illumination, the same optical property impinging on the CCD will cause fewer photons to strike the surface of the CCD for a package of lesser height than for a package of greater height.

The output of computer 36 is a height signal. In a preferred embodiment, the height signal is a two-bit digital signal. Thus, the value of the height signal indicates four ranges of height in the illumination zone. The height signal output is coupled to the input of the illumination compensation circuitry 42.

The output of initial gain adjustment 31 is a gain-adjusted analog reflected light intensity signal. The output of initial gain adjustment 31 is coupled to an input of analog-to-digital conversion circuitry 32. The output of analog-to-digital conversion circuitry 32 is a digital reflected light intensity signal. In a preferred embodiment, the digital reflected light intensity signal is an 8-bit signal. This output is coupled to an input of illumination compensation memory device 42.

The output of the position logic 40 through line 41 is coupled to an input of the illumination compensation memory device 42. This signal represents the variable horizontal position. In a preferred embodiment, this position logic output is a four-bit signal.

It will thus be understood that the illumination compensation memory device receives reflected light intensity value information along line 33. This information is provided as a series of individual reflected light intensity values. Each reflected light intensity value has an address defined by variable value information for that reflected light intensity value. In Figure 2, the values

of two variables define the address of each reflected light intensity value.

During calibration of an apparatus according to the invention, the reflected light intensity value information represents light reflected from a surface having a known, or reference, optical property. Such reflected light intensity value information is referred to as reference reflected light intensity value information.

During operation of an apparatus according to the invention, the reflected light intensity value information represents light reflected from a surface of unknown optical properties. Such reflected light intensity value information is referred to as operational reflected light intensity value information.

The initial gain adjustment apparatus will now be explained in greater detail. An 8-bit digital height signal is provided from the computer. The signal is input to a multiplier such as a CA 3338. The output of this logic unit is a gain-adjusted signal containing reflected light intensity value information. This is an analog signal. The amplitude of the output signal is determined by the height signal provided by the computer. In the preferred embodiment, a chip is provided that has 256 possible gain adjustments.

The height signal is generated by the computer depending on the height of the package. In one embodiment, a height sensor is employed, over a range of 28 inches, with a sensitivity of one-half inch. Thus, in that embodiment 56 different height readings are possible. In another embodiment, the height sensor is used over a range of 36 inches, thus making 72 different height readings possible. The computer has a look-up table stored in memory of gain signal values corresponding to each possible height value. The entries in the look-up table are obtained during calibration of the apparatus. The calibration of the apparatus is explained in detail below.

The output of the gain adjustment unit is coupled to an analog-to-digital conversion unit. The unit

may be, for example, a CA3318. This chip produces an 8-bit digital intensity signal.

It will be seen that the illumination compensation memory device has input lines dedicated to two types of input. A first type of input is uncompensated reflected light intensity data. This data is transmitted along line 33 from A/D conversion unit 32. The uncompensated reflected light intensity data is in the form of a series of intensity values.

The other type of input is variable value data. Each intensity value of uncompensated reflected light intensity data has an address defined by a value of each variable. Two input lines for variables are provided. One input line is for height information. The other variable input line is for horizontal position information. Thus each input intensity value is associated with a particular height value and horizontal position value.

The operation of the illumination compensation memory device will now be explained in more detail. The memory device maintains a set of look-up tables. The memory device uses the tables to remap input intensity values to compensated intensity values.

Each table corresponds to a particular combination of the variables. In a preferred embodiment, each table corresponds to a particular range of height and a particular range of horizontal location. The number of tables is dependent on the degree of detail desired in compensating. In a preferred embodiment, the illumination zone is divided into 32 segments of equal size across the width of the belt and 4 zones of equal size from the level of the belt to the top of the illumination zone. Consequently, 128 zones are provided. Thus, 128 tables are provided for adjusting the received intensity values. The table which is to be used for a given reading is determined by the height value and position value for the reading. Each table has a six-bit address. The height value is set for each entire label. The height signal provided to the illumination compensation memory device is accordingly a two-bit signal. The position value only

changes 16 times in every sweep of the CCD across the width of the belt. The position signal is provided to at least two memory devices corresponding to left and right sides of the illumination zone. In a preferred embodiment, there are four channels provided. Two channels receive reflected light intensity value information corresponding to the left half of the illumination zone. Two other channels receive reflected light intensity value information corresponding to the right side of the illumination zone. Thus the position value has 16 possible values, while 32 horizontal segments are provided. The same position signal corresponds to two position variable values, depending on which channel received the signal. The position value signal is thus a four-bit signal for each half of the illumination zone. Each time the position value changes, a new table is designated on each half of the illumination zone.

In the preferred embodiment, a CCD imager is employed having 4,096 pixels in a scan line. Thus each table remaps 128 reflected light intensity values.

The operation of the look-up tables in the memory devices is best understood with reference to Figure 8. Figure 8 is a graph of output versus input for three horizontal segments at a single height range. Figure 8 thus represents remapping of input to output for three of the 32 tables in a single height range.

Line 802, which is a staircase curve, represents the remapping of the input values to the output values by one table. Line 802 corresponds to segment 2, which remaps values of pixels 128-255. All input digital intensity values less than a black level threshold value are remapped to a black output level. In the illustration of Figure 8, the black output level is zero. The black level threshold value for line 802 is approximately 102. All input values above a white threshold are remapped to a white output level. For the segments illustrated in Figure 8, the white output level is the maximum output level of the memory device. The white threshold for line 802 is approximately 192.

Intensity values above the black level threshold and below the white level threshold are typically remapped according to a quantized linear function. Accordingly, an intensity value is remapped as follows. The output intensity value is equal to the sum of the black output level and the product of the absolute value of the difference between the black output level and the white output level by the quotient of the absolute value of the difference between the input value and the black level threshold divided by the absolute value of the difference between the white level threshold and the black level threshold. Remapping of the intensity values in this manner provides greater dynamic range for the data. Input values from a relatively compressed range between the black level threshold and the white level threshold are remapped to a wider range between the black level output and the white level output. For example, for segment 2, which corresponds to line 802, input intensity values from 105 through about 190, are remapped to values from 15 to 224.

The look-up tables for segments 8 and 14 are similar. The values of the black level threshold and the white level threshold are different for each segment. For example, for segment 9, represented by line 804, the black threshold value is about 118, and the white threshold value is about 220. For segment 14, represented by line 806, the black threshold value is about 126, and the white threshold value is about 230.

It should be noted that the remapping method of the invention is not confined to use with white and black threshold values. In fact, black is merely an example of a minimum reflected light intensity value. It is anticipated that the minimum reflected light intensity signal received by the memory device of the invention will correspond to a black surface. This is because it is anticipated that the system of the invention will be used with packages having labels printed with black areas. It is possible, in a different use of the invention, that a minimum reflected light intensity value would correspond to an optical property of the surface other than black.

In such event, a reference reflected light intensity value would be obtained using a calibration bar having areas with whatever optical property represented the minimum reflected light intensity value. This reference reflected light intensity value would be designated a threshold minimum reflected light intensity value. The memory means would store a look-up table designed to remap any operational reflected light intensity value equal to or less than the threshold minimum reflected light intensity value obtained from the reference to a minimum output level.

Similarly, it is possible to use the invention in a context in which there is a maximum reflective light intensity value that corresponds to an optical property of a surface other than white. In such a system, there would be a surface used for calibration that had the optical property corresponding to the maximum reflected light intensity value used for determining a reference maximum reflective light intensity value. This reference reflected light intensity value would be designated a threshold maximum reflected light intensity value. The memory device according to the invention would be programmed with a look-up table that remapped all operational reflected light intensity values equal to or greater than the threshold maximum reflected light intensity value to a maximum output level.

It will be seen that in the general case, any operational reflected light intensity value falling between the minimum reflected light intensity value threshold and the maximum light intensity value threshold would be remapped to an output value between a minimum output value and a maximum output value. For example, the values of operational reflected light intensity value falling between the minimum reflected light intensity value threshold and the maximum reflected light intensity value threshold can be remapped, to an output value between a minimum output value and a maximum output value. However, the intermediate values could be remapped according to any desired function.

As shown in Figure 2, the output values are transmitted by the memory device along an output line. The output signal is a series of compensated or remapped, reflected light intensity values. These may be four-bit values, with the four higher bits of an eight-bit output line used.

The process of calibration of the illumination compensation apparatus will now be described. It is necessary to carry out these calibration steps whenever changes are made that may affect the illumination. For example, replacement of a bulb or adjustment of a light requires that the illumination compensation apparatus be recalibrated. Recalibration includes reprogramming the memory device with tables for illumination compensation by remapping reflected light intensity values.

An overview of the calibration process may be understood with reference to Figure 4. The first step is adjustment of the attenuation gain, as shown by box 502 in Figure 4. The next step is the step of obtaining reference reflected light intensity values or profiles, shown by box 504 in Figure 4. The pattern of reflected light intensity values of any uniform optical property across the width of the zone is referred to as a "profile." The next step is the optional step of a hardware failure check, shown in box 506 in Figure 4. The next step is the step, required in a preferred embodiment of the invention, of combining certain reference reflected light intensity values, or reducing the number of profiles, shown in box 508 in Figure 4. The next step is the step of simplifying the reference reflected light intensity value profiles, shown in box 510 in Figure 4. The next step is the step of dividing the reference reflected light intensity values or profile data into segments, shown in box 512 on Figure 4. The next step is the step of finding the inverse of the reference reflected light intensity values or profile values, shown in box 514 in Figure 4. The next step is the step of multiplying the simplified and inverse simplified values to calculate a flat response, shown in box 515. The next step is the step of generating tables for

storage in a memory device. This is shown in box 516 in Figure 4. The last step is the step, shown in box 518 of Figure 4, of storing the tables in the memory devices.

In making these calibration adjustments, it is desirable to use a calibration bar 100 as shown in Figure 3. Alternating white strips 102 and black strips 104 are provided on the calibration bar 100 perpendicular to the length of the bar. For example, each strip may be 0.027 inches in width. The calibration bar 100 is sufficiently long that it can rest on pegs 82 of Figure 1, and extends across the entire belt. It will be seen that for obtaining the reference reflected light intensity values, a surface or surfaces must be provided having both the optical property corresponding to the minimum reflected light intensity value and the optical property corresponding to the maximum reflected light intensity value.

In the first step, shown as box 502, the attenuation gain is adjusted. The adjustments are made using an oscilloscope to monitor the levels. The goal in adjusting the attenuation gain is to assure that the black level is not below zero, and that the white levels are not saturated. The calibration bar is placed at various selected heights. The step of adjusting the gain is repeated at each selected height. The gain value for each selected height is stored in the computer for later processing. In a preferred embodiment, the selected heights are 1 inch, 4 inches, 8 inches, 12 inches, 16 inches, 20 inches, 24 inches and 28 inches above the conveyor belt. In a system for reading surfaces having heights up to 36 inches, additional readings would be taken.

The gain values set in the calibration process represent eight height points. In fact, in a preferred embodiment of the invention, there is provided height sensing means having the capability of detecting half-inch height intervals. The points intermediate the eight selected heights are obtained by standard curve-fitting techniques. In a preferred embodiment, the identical curve-fitting technique and software that is used as described below in the step of simplification of the com-

bined raw reflected light intensity data, or combined raw profiles, is employed. It may be seen, in the appended computer program listings, that the listing for this curve-fitting technique includes lines of code devoted to
5 dividing by four the gain values obtained so as to enable use of the same code for this curve-fitting operation.

The next step in calibration is the collection of data for forming the information required to accomplish illumination compensation. The collection of profile
10 data is preferably accomplished simultaneously with the step of gain adjustment. The readings given when the calibration bar 100 is placed across the illumination zone are recorded.

An image of the calibration bar is created at
15 each selected height. This image is raw, or uncompensated, data. The image of the calibration bar provides a reference image for the black level and the white level at a large number of horizontal positions. This raw image or raw profile, or reference reflected light intensity value
20 information, is transmitted to the computer. In the step of transmitting data to the computer, the hardware is configured so that the raw data is transmitted to the computer without compensation. In a preferred embodiment, each illumination compensation memory device is programmed
25 during calibration so that the data output by the illumination compensation memory devices is the same as the input. This is accomplished by removal of the memory devices, or PROM's, used in the prior run of the system with so-called "flat PROM's." These "flat PROM's" do not
30 adjust the data that is input. The output can be truncated before being stored. It may be truncated by a Pixel Bus (Trademark of Analogic, Peabody, Massachusetts) interface board, which is part of an embodiment of a computer system which receives data from the PROM's through a
35 buffer, and transmits data in truncated form to the remainder of the computer system.

As may best be seen with reference to Figure 5, the profile when viewing a calibration bar, or reference reflected light intensity value information, is substan-

tially in the form of two lines. With reference to Figure 5, the horizontal axis represents horizontal position across the illumination zone. The vertical axis represents reflected light intensity. In Figure 5, thirty-two data points, or pixels, have been chosen for display in the figure. The intensity values shown have been measured by placing an input of an oscilloscope on an intensity output line from the light-detecting device. The intensity value numbers are arbitrary.

10 The line designated by 602 in Figure 5 shows the white levels. As may be seen, there is substantial variation in the white levels. In particular, the white levels drop on the edges of the illumination zone. The line designated by 604 represents the black levels. As may be
15 seen, there is also considerable variation in the black levels. A profile similar to Figure 5 is obtained for each height interval at which the calibration bar is placed. The set of raw profiles are stored for further processing. Each raw profile is stored in a separate file
20 in the computer.

 After collection and storing of the reference intensity values, there is an optional step of checking the reference intensity value information data for evidence of hardware failure. A hardware failure in the
25 system would make reading of labels with a reasonable degree of accuracy impossible. Such hardware failures include a failure of a portion of the illumination apparatus, such as a bulb, or of a portion of the light sensing means, such as a failure of a CCD array.

30 Checking the profile data for evidence of hardware failures is accomplished by computing a variance and standard deviation. The equation for the variance, $\text{var}(X_{1..n})$, is the following:

$$\text{var}(x_1 \dots x_n) = \frac{1}{N-1} \times \sum_{j=1}^N \left(x_j - \left(\frac{1}{N} \times \sum_{j=1}^N x_j \right) \right)^2$$

where X is the intensity value and N is the number of
35 pixels in a profile. The standard deviation $\sigma(x_{1..n})$ is defined by the equation of $\sigma(x_{1..n}) = \sqrt{\text{var}(x_{1..n})}$

In a preferred embodiment, a routine to calculate the variance and standard deviation is taken from W. Press, et al., "Numerical Recipes in C", (Cambridge University Press 1988) at page 475. The listing for the software routine
5 called "moment.c" is shown in the appended source code listings.

The next step after the optional step of checking the raw profile data for hardware errors is the reduction of the number of profiles obtained in the data collection step to the number of profiles actually used
10 during illumination compensation. For example, in a preferred embodiment, in the data collection step, eight profiles are collected and stored. In a preferred embodiment, the illumination zone is divided into only four
15 vertical divisions. Consequently, the eight profiles must be combined into four. It has been found desirable to take readings at levels 1 inch, 4 inches, 8 inches, 12 inches, 16 inches, 20 inches, 24 inches and 28 inches above the belt. It has been found desirable to use the
20 highest and lowest profiles individually. This is because these profiles are the extremes of lighting, and usually quite different from an average of intermediate height profiles. In this embodiment, the 1 inch and 28 inch profiles are used without adjustment. The 4 inch, 8 inch
25 and 12 inch profiles are averaged on a pixel-by-pixel basis. The 16 inch, 20 inch and 24 inch profiles are also averaged on a pixel-by-pixel basis. The result is four combined profiles. This is referred to as the combined reference reflected light intensity value information.

30 The next step is the step of simplifying the reference intensity value data. This is accomplished in a preferred embodiment by applying curve-fitting techniques to the combined raw profiles. A variety of curve-fitting algorithms may be used. In a preferred embodiment, a
35 second-degree polynomial curve-fit algorithm is used. The procedure to convert raw profile data is as follows. The data is represented on a graph. The y-axis of the graph corresponds to reflected light intensity value and the x-axis corresponds to a horizontal position in the illumina-

tion zone. The equation for finding the fitted curve from the raw profiles is the following:

$$\begin{bmatrix} m & \sum x & \sum x^2 \\ \sum x & \sum x^2 & \sum x^3 \\ \sum x^2 & \sum x^3 & \sum x^4 \end{bmatrix} \times \begin{bmatrix} b_0 \\ b_1 \\ b_2 \end{bmatrix} = \begin{bmatrix} \sum y \\ \sum x \cdot y \\ \sum x^2 \cdot y \end{bmatrix}$$

The source code for this is shown in CURVE-FIT C, which is included herewith in the appended software listings.

Referring to Figure 6, there is shown a flow chart depicting the steps of the process of data simplification employing a preferred curve-fitting technique. In the step illustrated by the box labeled 702 on Figure 6, there is shown the step of allocating arrays. The allocation of arrays is a conventional preliminary step to performing vector and matrix mathematics. The next step is the step of inserting the X and Y values in the matrices and arrays. The X values are the pixel numbers, which correspond to horizontal position in the illumination zone. The Y values are reflected light intensity values. This step is shown at box 704 in Figure 6. The next step, shown in box 706 in Figure 6, is the step of computing the X summations. The X summations are elements of the left hand matrix in the above equation for the curve-fit. This step requires computation of summations of an array of X-values raised to various powers. A preferred software subroutine for this computation is shown under the file name "sum_a" in the appended source code listings. The next step in the process is the step, shown in box 708 in Figure 6, of computing the XY summations. This step requires taking two arrays of data, the array of X-values and the array of Y-values, and calculating a summation. The values in the array of Y-values are raised to increasing powers. A software subroutine for this step is shown under the file name "sum_ab" in the appended source code listings. The next step, shown at box 710, is transferring the values obtained from the step of computing the X summations to a matrix. This is the matrix shown on the left hand side of the above equation for the second-degree

polynomial curve-fit. As may be seen from inspection of the above equation, it is necessary to rearrange the equation to solve for the vector (b_0, b_1, b_2) . Thus, it is required to find the inverse of the matrix of X summations. This step is performed in a preferred embodiment by first finding the inverse of the matrix of X summations on a column-by-column basis. This is done by computing the LU decomposition of a row-wise permutation of the matrix. A preferred function for finding the LU decomposition is found in the above-mentioned "Numerical Recipes in C" at Chapter 2, page 43. The source code for a software routine is listed under file name "ludcmp.c" in the appended source code listings. This step is shown at box 712 in Figure 6. With reference to Figure 6, it will be seen that the next step is solving the resulting linear equations shown in box 714. A function for solving such linear equations is disclosed in the above-mentioned "Numerical Recipes in C", chapter 2, page 22. The source code for a software routine for this function is listed under the filename "lubksb.c" in the appended source code listings. The next step after solution of the linear equations is calculating the coefficients of B, as shown in box 716 in Figure 6. Once this step is complete, a set of appropriate curves has been derived.

The result of the step of simplifying data using a second-degree polynomial curve-fit is illustrated in Figure 7. The graph shown in Figure 7 is similar to the graph shown in Figure 5. Thus, the horizontal axis of Figure 7 represents the position horizontally across the illumination zone. The vertical axis represents a value of intensity of reflected light. Line 802 near the top of the graph represents the reflected light intensity data, after the step of data simplification, for the white strips on the calibration bar. Line 804, extending across near the bottom of the graph, represents reflected light intensity values, after data simplification, for the black strips on the calibration bar.

After the step of processing the raw profile data into curve-fitted profile data, there are four sets of profile data. Each set of profile data represents a profile corresponding to one of four ranges of height values in the illumination zone. Each set of profile data is next broken into 32 segments as shown by box 512 in Figure 4. The 32 segments represent the 32 horizontal segments of the illumination zone. Each of the 32 segments corresponds to 128 pixels across the illumination zone.

The simplified reference reflected light intensity values, or simplified profile data, as shown by box 514 in Figure 4, is then mapped to an inverse curve. In obtaining the inverse, each line is first rescaled by conventional techniques.

The values contained in the inverse curve are used in programming the illumination compensation memory devices. The values must be placed in a format that can be accepted by the memory devices. In a preferred embodiment, an Intel MCS-86 Hex file is created for each of the PROM's. The data stored is in the form of a table having 256 elements.

The data is then converted to a form which is compatible with the various channels into which the data is divided by the CCD units. For example, in a preferred embodiment, there are four channels. These four channels represent respectively odd and even pixels on the left and right sides of the belt. Accordingly, the data generated is divided into appropriate segments for each channel. Once the data is divided in this manner, the data is stored in computer memory in four parts. The data is transferred from the computer to the memory device. In a preferred embodiment, the memory device is an ultra-violet light erasable programmable read-only memory (PROM). A Cypress 7C254 may be used, for example. In order to transfer data from the computer to such a device, a PROM burner is provided. The input of the PROM burner is coupled to a port of the computer so that data may be transferred from the computer to the PROM burner. In

accordance with usual techniques, the PROM burner is connected to the PROM so as to burn the data into the PROM. Once the data has been stored in the memory device, the memory device is then inserted into the appropriate
5 board. The process of burning and inserting the PROM is repeated for each channel. When the memory devices have been programmed and are in place in the circuit, the apparatus of the invention may be operated.

The look-up tables may also be maintained in a
10 random access memory. Such a memory would be programmable from a system computer.

It will be appreciated that there are considerable variations that can be accomplished in a method and apparatus according to the invention without departing
15 from its scope. As a result, although a preferred embodiment of a method and apparatus of the invention has been described above, it is emphasized that the invention is not limited to a preferred embodiment and that there exist other alternative embodiments that are fully encompassed
20 within the invention's scope, which is intended to be limited only by the scope of the appended claims.


```

/*****
* File Name: moment.c
*
* File Function: Routine to compute the average, standard deviation
*                and variance of the data passed.
*
* Notes: Function was taken from Numerical Recipe in 'C' see *
*        chapter 13 pg 475
*        this function is a subset of the moment function.
*
* History: 1988 Created
*
*****/

#include <math.h>

void moment(data,m,n,index,ave,sdev,svar)
int m,n,index;
float **data,*ave,*sdev,*svar;
{
    int j;
    float s;
    void nrerror();

    if (n <= 1)
        nrerror("n must be at least 2 in MOMENT");

    s=0.0;
    for (j = 1; j <= n;j++)
        s += data[index][j];

    *ave=s/n;

    *svar = 0.0;

    for(j = 1; j <= n; j++)
        *svar += fabs(s = data[index][j] - (*ave));

    *svar /= n;
    *sdev=sqrt(*svar);
}

```

```

/*****
*
* @Copyright 1991 United Parcel Service of America, Inc.
* All Rights Reserved.
*
* Confidential, Unpublished Property of United Parcel Service
* Use and Distribution Limited Solely to Authorized Personnel.
*
* The use, disclosure, reproduction, modification, transfer,
* or transmittal of this work for any purpose in any form or
* by any means without the written permission of United Parcel
* Service is strictly prohibited.
*
*****/

```

```

/*****
* File Name: curvefit.c
*
* File Function: To take the profile or sa gain values and apply
*                a second degree curve fit to the data. The
*                function is controlled by the variable type
*                which selects arrays size for the data inputs
*                again profile data or sa gain data.
*                The output of the function is the three
*                coefficients that best fit the data.
*
* Notes:
*
* History: JSK 4-June-91 Created
*          JSK 20-June-91 Fix x index for SA data
*
*****/

```

```

#include <stdio.h>
#include <math.h>
#include "nr.h"
#include "nrutil.h"

```

```

#define VECMAX 32
#define ARRAMEX 3
#define SUMAX 4

```

```

float sum_a();
float sum_ab();

```

```

curvefit(data,type,index,b)
float **data,*b;
int index;
char type;
{
    float *x,*y,**a,*xsum,*xysum,*col,**ai,d,power;
    int i,j,k,m,*indx;
    FILE *fp;

    x = vector(1,VECMAX);          /* allocate array space */
    y = vector(1,VECMAX);
    xsum = vector(1,VECMAX);
    xysum = vector(1,VECMAX);
    a = matrix(1,ARRAMAX,1,ARRAMAX);
    ai = matrix(1,ARRAMAX,1,ARRAMAX);
    col = vector(1,VECMAX);
    indx = ivector(1,VECMAX);

    if(type == 'P')                /* Dealing with profile gains */
        m = VECMAX;
    else
        m = VECMAX/4;              /* Dealing with SA gains */

    if(type == 'P')                /* Profile Gains */
    {
        for(i = 1; i <= m; i++) /* Get x values */
            x[i] = i;
    }
    else                            /* Sa gains */
    {
        for(i = 1, j = 1; i <= m; i++, j += 4)
            x[i] = j;              /* Note Sa gains are increments 4 */
    }

    for(i = 1; i <= m; i++)         /* Get y values */
        y[i] = data[index][i];

    power = 1.0;                   /* Compute the x summations */
    for(i = 1; i <= SUMAX; i++)
        xsum[i] = sum_a(x,m,power++);

    power = 1.0;                   /* Compute the xy summations */
    xysum[1] = sum_a(y,m,power);

    for(i = 2; i < SUMAX; i++)
        xysum[i] = sum_ab(x,y,m,power++);
}

```

```

a[1][1] = (float) m;          /* Fill in the A matrix */
a[1][2] = xsum[1];
a[1][3] = xsum[2];
a[2][1] = xsum[1];
a[2][2] = xsum[2];
a[2][3] = xsum[3];
a[3][1] = xsum[2];
a[3][2] = xsum[3];
a[3][3] = xsum[4];

                                /* find inverse column by column */
ludcmp(a,ARRAMAX,indx,&d);      /* perform LU decomposition */
for(j = 1; j <= ARRAMAX; j++) /* Math functions ludcmp() & */
{                               /* lubksb() obtain from */
    for(i = 1; i <= ARRAMAX; i++) /* Numerical Recipes */
        col[i] = 0.0;
    col[j] = 1.0;
    lubksb(a,ARRAMAX,indx,col); /* Solve n linear equations
    for(i = 1; i <= ARRAMAX; i++) /* A x X = B */
        ai[i][j] = col[i];
}

for(i = 1; i <= ARRAMAX; i++) /* Now find coefficients of b */
{
    b[i] = 0.0;
    for(j = 1; j <= ARRAMAX; j++) /* b = A inverse * y */
        b[i] += ai[i][j] * xysum[j];
}

free_ivector(indx,1,VECMAX);    /* Deallocate space */
free_vector(col,1,VECMAX);
free_matrix(ai,1,ARRAMAX,1,ARRAMAX);
free_matrix(a,1,ARRAMAX,1,ARRAMAX);
free_vector(xysum,1,VECMAX);
free_vector(xsum,1,VECMAX);
free_vector(y,1,VECMAX);
free_vector(x,1,VECMAX);
}

```

```

/*****
* File Name: sum_a
*
* File Function: To take an array of data and find a summation *
                  raised to a variable power
*
* Notes:
*
* History: JSK 4-June-91 Created
*
*****/

```

```

float sum_a(a,limit,power)
float *a,power;
int limit;
{
    int i;
    float sum = 0.0;

    for(i = 1; i <= limit; i++)
        sum += pow(a[i],power);

    return sum;
}

```

```

/*****
* File Name: sum_ab
*
* File Function: To take two arrays of data and find a summation
                  raised to variable power
*
* Notes:
*
* History: JSK 4-June-91 Created
*
*****/

```

```

float sum_ab(a,b,limit,power)
float *a,*b,power;
int limit;
{
    int i;
    float sum = 0.0;

    for(i = 1; i <= limit; i++)
        sum += (pow(a[i],power) * b[i]);

    return sum;
}

```

```

/*****
* File Name: lubksb.c
*
* File Function: Solves the set of n linear equations  $A \cdot X = b$ 
* Notes: Function was taken from Numerical Recipe in 'C' see .
*         chapter 2 pg 44
* History: 1988 Created
*
*****/

```

```

void lubksb(a,n,indx,b)
float **a,b[];
int n,*indx;
{
    int i,ii=0,ip,j;
    float sum;

    for(i = 1;i <= n;i++)
    {
        ip = indx[i];
        sum = b[ip];
        b[ip] = b[i];
        if(ii)
            for(j = ii;j <= i-1;j++)
                sum -= a[i][j]*b[j];
        else
            if(sum)
                ii = i;
        b[i] = sum;
    }
    for(i = n;i >= 1;i--)
    {
        sum = b[i];
        for(j = i+1;j <= n;j++)
            sum -= a[i][j]*b[j];
        b[i] = sum/a[i][i];
    }
}

```

```

/*****
* File Name: ludcmp.c
*
* File Function: Given an nxn matrix routine replaces it by the LU
*                decomposition of a row wise permutation of *
*                itself.
*
* Notes: Function was taken from Numerical Recipe in 'C' see *
*        chapter 2 pg 43
* History: 1988 Created
*
*****/

#include <math.h>

#define TINY 1.0e-20;

void ludcmp(a,n,indx,d)
int n,*indx;
float **a,*d;
{
    int i,imax,j,k;
    float big,dum,sum,temp;
    float *vv,*vector();
    void nrerror(),free_vector();

    vv=vector(1,n);
    *d=1.0;
    for (i=1;i<=n;i++) {
        big=0.0;
        for (j=1;j<=n;j++)
            if ((temp=fabs(a[i][j])) > big) big=temp;
        if (big == 0.0) nrerror("Singular matrix in routine
LUDCMP");
        vv[i]=1.0/big;
    }
    for (j=1;j<=n;j++) {
        for (i=1;i<j;i++) {
            sum=a[i][j];
            for (k=1;k<i;k++) sum -= a[i][k]*a[k][j];
            a[i][j]=sum;
        }
        big=0.0;
        for (i=j;i<=n;i++) {
            sum=a[i][j];
            for (k=1;k<j;k++)
                sum -= a[i][k]*a[k][j];
            a[i][j]=sum;
            if ( (dum=vv[i]*fabs(sum)) >= big) {
                big=dum;
                imax=i;
            }
        }
    }
}

```

```

    if (j != imax) {
        for (k=1;k<=n;k++) {
            dum=a[imax][k];
            a[imax][k]=a[j][k];
            a[j][k]=dum;
        }
        *d = -(*d);
        vv[imax]=vv[j];
    }
    indx[j]=imax;
    if (a[j][j] == 0.0) a[j][j]=TINY;
    if (j != n) {
        dum=1.0/(a[j][j]);
        for (i=j+1;i<=n;i++) a[i][j] *= dum;
    }
}
free_vector(vv,1,n);
}

#undef TINY

```



```

/*
 * File: genprom.c
 * Auth: Robert K Pekarek
 * Date: 03 Jan 1990
 *
 * Copyright (c) 1990 United Parcel Service.
 *
 * Purpose: Generate prom object files for SA correction tables.
 *
 * Edit: Jim S Kunicki (JSK) added #define min marco to compile
 *       under Unix
 *       Changed program to be a callable function
 */

#define HALVES 2
#define FIRST 0
#define SECOND 1
#define HZ 4
#define PZ 16
#define DATABITS 256
#define SIZE 16
#define VERSION 1
#define REVISION 0
#define min(a,b) (((a) < (b)) ? (a) : (b))

#include <stdlib.h>
#include <stdio.h>
#include <string.h>
#include <ctype.h>
#include <math.h>
#ifdef MSDOS
#include <process.h>
#endif

/*****
extern void main(int argc, char **argv);
static FILE *GetInputFile(int argCount, char **stringVector);
static void GetFSOpsRND(FILE *fp, int *full, float *preScale, int
*quantize,
    int *inputScale);
static void BuildRawOutOffset(FILE *fp, float *off1, float *raw1,
                                float *off2,
float *raw2);
static void CalcProm(float *off1, float *raw1,
    int full, float preScale, int quantize,
int inputScale,
    unsigned char *result, float *scale);
static void WriteIntelHex(unsigned char *result, int aLen, char
*outFileName);
static void WriteText(unsigned char *result, char *outFileName);
static void WriteScaleFactors(float *scale, int aLen, char
*outFileName);
*****/

```

```

static FILE *GetInputFile();
static void BuildRawOutOffset();
static void CalcProm();
static void WriteIntelHex();
static void WriteText();
static void WriteScaleFactors();
static void GetFSOpsRND();

/* Global arrays */
float rawOut[HALVES][HZ][PZ], offset[HALVES][HZ][PZ];
float preScale, scale[HALVES][HZ][PZ];
unsigned char results[HALVES][HZ][PZ][DATABITS];

/*
 * main - get info and build prom files
 */
/*****/
genprom()
{
FILE *fp;
int fullScaleOutput, roundToNear, inpScale;

    fprintf(stderr, "Genprom version %li.%li\n", VERSION,
REVISION);

    /* Open file for input */
    if ((fp = GetInputFile()) == NULL)
    {
        fprintf(stderr, "Usage: <required file name>\n");
        return;
    }

    /* Get full scale output, pre-scale, and quantize factor */
    GetFSOpsRND(fp, &fullScaleOutput, &preScale, &roundToNear,
&inpScale);

    /* Build input offset and raw data arrays */
    BuildRawOutOffset(fp, &offset[FIRST][0][0], &rawOut[FIRST][0][0],
&offset[SECOND][0][0], &rawOut[SECOND][0][0]);

    /* Close input file */
    fclose(fp);

    /* Calculate scale factors and results */
    CalcProm(&offset[FIRST][0][0], &rawOut[FIRST][0][0],
fullScaleOutput, preScale, roundToNear, inpScale,
&results[FIRST][0][0][0], &scale[FIRST][0][0]);
    CalcProm(&offset[SECOND][0][0], &rawOut[SECOND][0][0],
fullScaleOutput, preScale, roundToNear, inpScale,
&results[SECOND][0][0][0], &scale[SECOND][0][0]);

```

```

        /* Write prom data to files */
        WriteIntelHex(&results[FIRST][0][0][0],      PZ*HZ*DATABITS,
"AB.PRM");
        WriteIntelHex(&results[SECOND][0][0][0],    PZ*HZ*DATABITS,
"CD.PRM");

        /* Write prom data to files */
        WriteText(&results[FIRST][0][0][0], "AB.TXT");
        WriteText(&results[SECOND][0][0][0], "CD.TXT");

        /* Write out scale factors */
        WriteScaleFactors(&scale[FIRST][0][0],      PZ*HZ*HALVES,
"SF.TXT");

        /* Acknowledge completion */
        fprintf(stderr, "Done generating prom files and scale
factors\n");
    }

    /*
     * GetInputFile - verify argument count and retrieve file name
     */
    /* ----- */
    static FILE *GetInputFile()
    {
        FILE *tf;

        /* Return invalid argument count */
        /******
        if (argCount != 2)
        {
            return NULL;
        }
        *****/

        /* Open file note hard coded for now */
        if ((tf = fopen("DATA.OUT", "rt")) == NULL)
        {
            perror("");
        }

        return tf;
    }

    /*
     * GetFSOpsRND - read scalers from input file in designed format
     */
    /* ----- */

#ifdef SPARC
#define MAX_PATH 144      /* Note the was a MICROSOFT "C" define
within */
#endif
/* stdlib.h, SUN "C" is not supported */

```

```

static void GetFSOpsRND(fp,full,preScale,quantize,inputScale)
FILE *fp;
int *full,*quantize,*inputScale;
float *preScale;
{
    char buffer[_MAX_PATH];
    int rc;

    /* Get preScale from first line */
    if (fgets(buffer, _MAX_PATH, fp) == NULL)
    {
        perror("");
        exit(1);
    }
    if ((rc = sscanf(buffer, "PS = %f", preScale)) != 1)
    {
        perror("");
        exit(1);
    }

    /* Get full scale output from second line */
    if (fgets(buffer, _MAX_PATH, fp) == NULL)
    {
        perror("");
        exit(1);
    }
    if ((rc = sscanf(buffer, "FSO = %i", full)) != 1)
    {
        perror("");
        exit(1);
    }

    /* Get round to nearest from third line */
    if (fgets(buffer, _MAX_PATH, fp) == NULL)
    {
        perror("");
        exit(1);
    }
    if ((rc = sscanf(buffer, "RND = %i", quantize)) != 1)
    {
        perror("");
        exit(1);
    }

    /* Get input scaling from fourth line */
    if (fgets(buffer, _MAX_PATH, fp) == NULL)
    {
        perror("");
        exit(1);
    }
    if ((rc = sscanf(buffer, "IS = %i", inputScale)) != 1)
    {

```

```

        perror("");
        exit(1);
    }
}

/*
 * BuildRawOutOffset - read raw outputs and offsets from input file
 */
/* ----- */
static void BuildRawOutOffset(fp, off1, raw1, off2, raw2)
    FILE *fp;
    float *off1, *raw1, *off2, *raw2;
{
#define Numparms 2
#define Longbuffer 300
#define Startpos 23
    static char MatchString[] = "Diag";

    char buffer[Longbuffer], *rc, *bufferPtr;
    float *ptr1, *ptr2;
    int hz, pz, count, bufferOffset;

    /* Find first line containing the identifying string */
    do
    {
        rc = fgets(buffer, Longbuffer, fp);

        while ((rc != NULL) && strncmp(buffer, MatchString,
            strlen(MatchString)));

        if (rc == NULL)
        {
            perror("No data found");
            exit(1);
        }

        /* For all height zones in the file, and not at end of file
        */
        for (hz = 0; (hz < HZ) && (rc != NULL); ++hz)
        {
            /* For both sets of input data */
            for (ptr1 = off1, ptr2 = off2 + PZ - 1, count = 0;
                count < Numparms;
                ptr1 = raw1, ptr2 = raw2 + PZ - 1, ++count)
            {
                /* Locate beginning of data on line */
                bufferPtr = &buffer[Startpos];

                /* Get the data for the left side */
                for (pz = 0; pz < PZ; ++pz, ++ptr1)
                {
                    sscanf(bufferPtr, "    %f,%n", ptr1,

```

```

&bufferOffset);
        bufferPtr += bufferOffset;
    }

    /* Get the data for the right side */
    for (pz = 0; pz < PZ; ++pz, --ptr2)
    {
        sscanf(bufferPtr, "    %f,%n", ptr2,
&bufferOffset);
        bufferPtr += bufferOffset;
    }

    /* Get next line */
    rc = fgets(buffer, LONGBUFFER, fp);
}

/* Increment data pointers */
off1 += PZ;
off2 += PZ;
raw1 += PZ;
raw2 += PZ;
}

}

/*
 * CalcProm - compute prom index values from given empirical data
 */
/* ----- */
s      t      a      t      i      c      v      o      i      d
CalcProm(off1,raw1,full,preScale,quantize,inputScale,result,scale)
float *off1,*raw1,preScale,*scale;
int full,quantize,inputScale;
unsigned char *result;
{
int outerLoop, sel;
float tempScale, tempStore;

/* For each of the height and position zones */
for (outerLoop = 0; outerLoop < HZ * PZ; ++outerLoop)
{

    /* Compute and store the scale value */
    tempScale = *scale++ =
        (full / inputScale) / (preScale * *raw1 - *off1);

    /* For each data selector */
    for (sel = 0; sel < DATABITS; ++sel)
    {

        /* Compute the desired output and store to the
result area */
        tempStore = sel - *off1 * inputScale;
        if (tempStore < (float)0) tempStore = (float)0;
        tempStore *= tempScale;
    }
}

```

```

        if (tempStore > (float)full) tempStore =
(float)full;
        tempStore = (float)floor(tempStore /
(float)quantize + (float)0.5) *
(float)quantize;
        *result++ = (unsigned char)tempStore;
    }

    /* Increment pointers */
    raw1++;
    off1++;
}

/*
 * WriteIntelHex - put character data out to given file in the
intel hex
 *
 * format
 */
/* ----- */
static void WriteIntelHex(result, aLen, outFileNames)
unsigned char *result;
int aLen;
char *outFileNames;
{
FILE *fp;
int count, tempLen, chkSum, dataCount;

    /* Open file */
    fp = fopen(outFileNames, "wt");

    /* For each data group of SIZE */
    count = 0;
    while (aLen > 0)
    {
        /* Put out intel Hex line */
        tempLen = min(SIZE, aLen);
        chkSum = 0x7000 - tempLen - (count / DATABITS) - (count
% DATABITS);
        fprintf(fp, ":%02X%04X00", tempLen, count);

        /* Put out data */
        for (dataCount = 0; dataCount < tempLen; ++dataCount)
        {
            fprintf(fp, "%02X", *result);
            chkSum -= *result++;
        }

        /* Put out checksum */
        fprintf(fp, "%02X\n", chkSum % DATABITS);

        aLen -= tempLen;
        count += tempLen;
    }
}

```

```

    }

    /* Put out end record */
    fprintf(fp, ":00000001FF\n");

    fclose(fp);
}

/*
 * WriteText - put character data out to given file in text format
 */
/* ----- */
static void WriteText(result,outFileName)
    unsigned char *result;
    char *outFileName;
{
    FILE *fp;
    int hz, pz, dataCount;

    /* Open file */
    fp = fopen(outFileName, "wt");

    /* For each data group */
    for (hz = 0; hz < HZ; ++hz) {
        for (pz = 0; pz < PZ; ++ pz) {
            fprintf(fp, "\n      HZ = %1i      PZ = %2i\n", hz, pz);

            /* Put out data */
            for (dataCount = 0; dataCount < DATABITS; ++dataCount)
            {
                fprintf(fp, "% 3u ", *result++);
                if (!(dataCount + 1) % SIZE) fprintf(fp, "\n");
            }
        }
    }

    fclose(fp);
}

/*
 * WriteScaleFactors - put scale factors to given file in ASCII
 * format
 */
/* ----- */
static void WriteScaleFactors(scale,aLen,outFileName)
    float *scale;
    int aLen;
    char *outFileName;
{
    FILE *fp;
    int count;

    /* Open file */
    fp = fopen(outFileName, "wt");

```



```

    fprintf(fp, "Channels A/B\n");

    /* Write all scale factors */
    for (count = 0; count < aLen/2; ++count)
    {
        fprintf(fp, "%2i %2i %4.1f\n", count / PZ, count % PZ,
*scale++);
    }

    fprintf(fp, "Channels C/D\n");
    for (count = 0; count < aLen/2; ++count)
    {
        fprintf(fp, "%2i %2i %4.1f\n", count / PZ, count % PZ,
*scale++);
    }

    fclose(fp);
}

```

What is claimed is:

1. A method for illumination compensation for use in a system for reading surfaces having at least black and white areas, comprising the steps of:

- 5 (a) storing reference reflected light intensity information corresponding to at least black and white areas in the form of a plurality of intensity values, each one of said intensity values having an address defined by a value of at least one variable;
- 10 (b) calculating from said reference reflected light intensity information, a plurality of black threshold values and a plurality of white threshold values, each one of said threshold values associated with selected values of each of said at least one variable;
- 15 (c) receiving operational reflected light intensity information in the form of a plurality of operational reflected light intensity values, each one of said operational reflected light intensity values having an address defined by a value of each of said at least one variable;
- 20 (d) remapping each of said operational reflected light intensity values to one of a plurality of output compensated reflected light intensity values, including the steps of remapping any of said operational reflected light intensity values equal to or less than
25 said black threshold value for its address to a black level output and remapping any of said operational reflected light intensity values equal to or greater than said white threshold value for its address to a white level output; and

- 30 (e) producing an output signal comprising said output compensated reflected light intensity values.

2. A method as recited in claim 1, wherein said step of remapping further includes remapping any of said operational light intensity values less than said white threshold value for its address and greater than said black threshold value for its address to an output
5 value equal to the absolute value of the difference be-

tween the white output level and the black output level, multiplied by the quotient of said operational light intensity value less the black level threshold divided by the absolute value of the difference between the black
10 level threshold and the white level threshold.

3. A method as recited in claim 1, wherein said black output level is zero, and said white output level is a maximum output level.

4. A method as recited in claim 1, wherein said step of storing comprises storing reference reflected light intensity information corresponding to at least black and white areas in a plurality of profiles, each one
5 of said profiles displaying said reference reflected light intensity information plotted against one of said variables, each other of said variables being a selected constant for each profile.

5. A method for illumination compensation in a system for processing of signals representing reflected light intensities from a surface, comprising the steps of:

(a) receiving reflected light intensity
5 information in the form of a plurality of intensity values;

(b) receiving, associated with each one of said reflected light intensity values, a value of at least one variable;

10 (c) remapping each of said reflected light intensity values to one of a plurality of compensated reflected light intensity values, including the steps of remapping any of said reflected light intensity values equal to or less than a preselected threshold minimum
15 reflected light intensity value associated with the value of each of said variables for said reflected light intensity value to a minimum level output and remapping any of said reflected light intensity values equal to or greater than a preselected threshold maximum reflected light
20 intensity value, associated with the value of each of said variables for each reflected light intensity value, to a maximum level output.

6. A method as recited in claim 5, wherein said step of remapping further includes remapping any of said reflected light intensity values greater than said preselected threshold minimum reflective light intensity value associated with the value of each of said variables for each reflective light intensity value, and less than said preselected threshold maximum reflective light intensity value associated with the value of each of said variables for each reflected light intensity value, to an output value equal to the absolute value of the difference between the maximum output level and the minimum output level, multiplied by the quotient of the reflective light intensity value less the threshold minimum light intensity value divided by the absolute value of the difference between the preselected threshold maximum reflective light intensity value and the preselected threshold minimum light intensity value.

7. A method as recited in claim 6, wherein said minimum level output is zero and said maximum level output is a maximum possible output level.

8. An apparatus for illumination compensation in a system for processing of signals representing reflected light intensities from a surface, comprising:

(a) means for receiving reflected light intensity information in the form of a plurality of intensity values;

(b) means for receiving, associated with each one of said reflected light intensity values, a value of at least one variable;

(c) memory means for remapping each of said reflected light intensity values to one of a plurality of compensated reflected light intensity values, including memory means for remapping any of said reflected light intensity values equal to or less than a preselected threshold minimum reflected light intensity value associated with the value of each of said variables for said reflected light intensity value to a minimum level output and memory means for remapping any of said reflected light intensity values equal to or greater than a preselected

20 threshold maximum reflected light intensity value associated with the value of each of said variables for each reflected light intensity value, to a maximum level output.

9. An apparatus as recited in claim 8, further comprising memory means for remapping any of said reflected light intensity values less than said preselected threshold maximum light intensity value associated with
5 the value of said variables for said reflected light intensity value and greater than said preselected threshold minimum reflected light intensity value corresponding to a value of each of said variables for said reflected
10 light intensity value, linearly to a value greater than said minimum level output and less than said maximum level output.

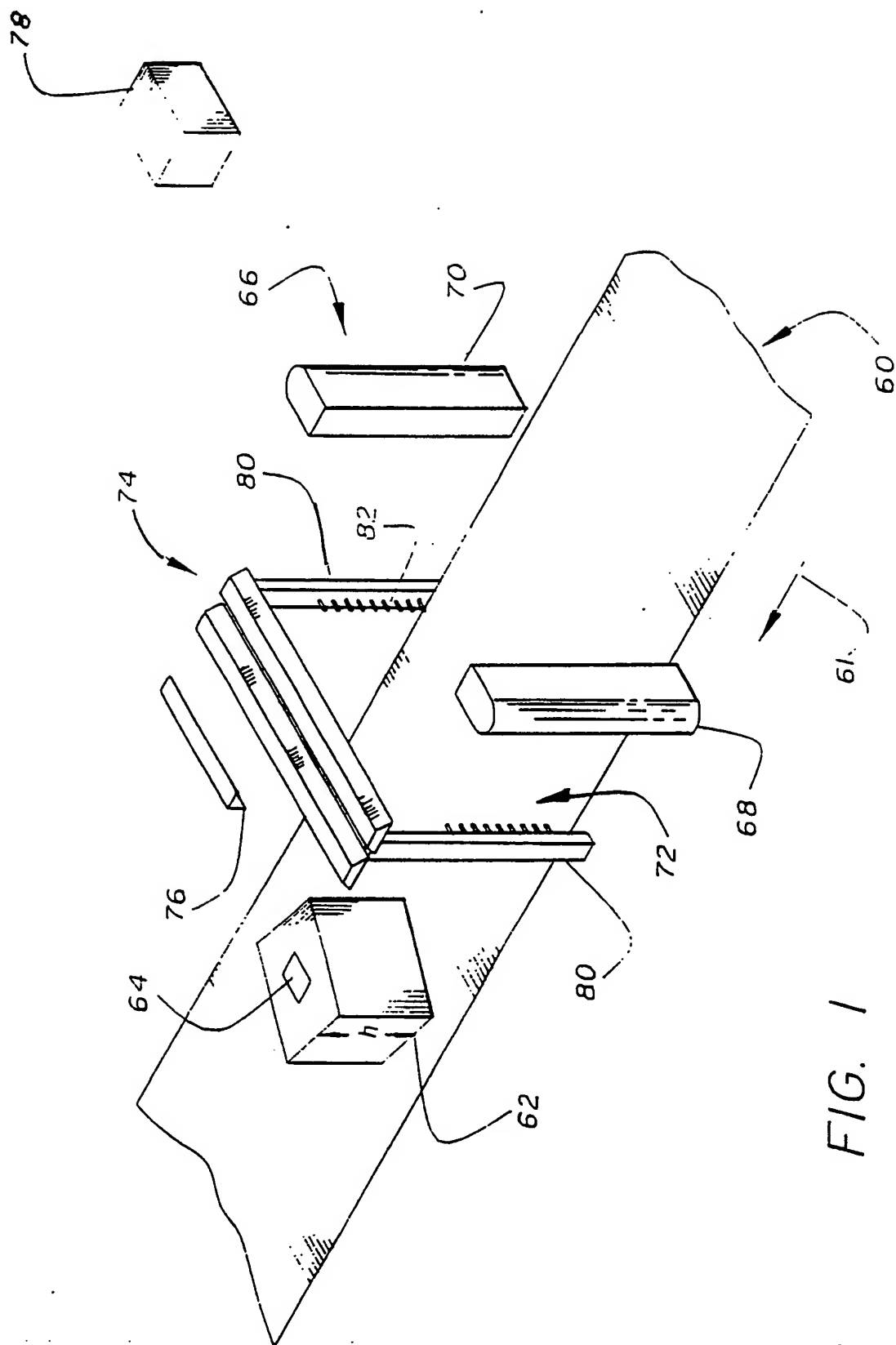


FIG. 1

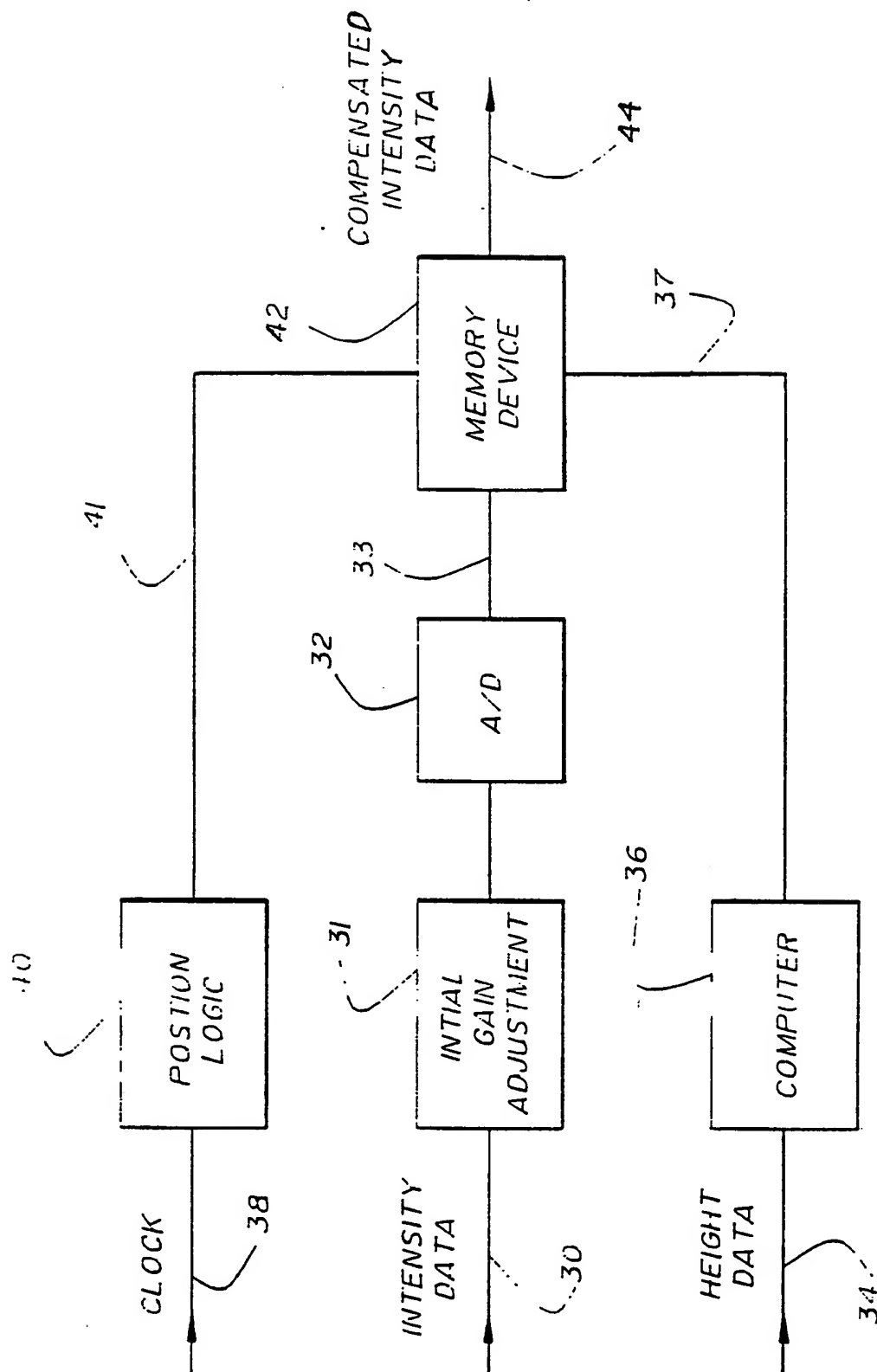
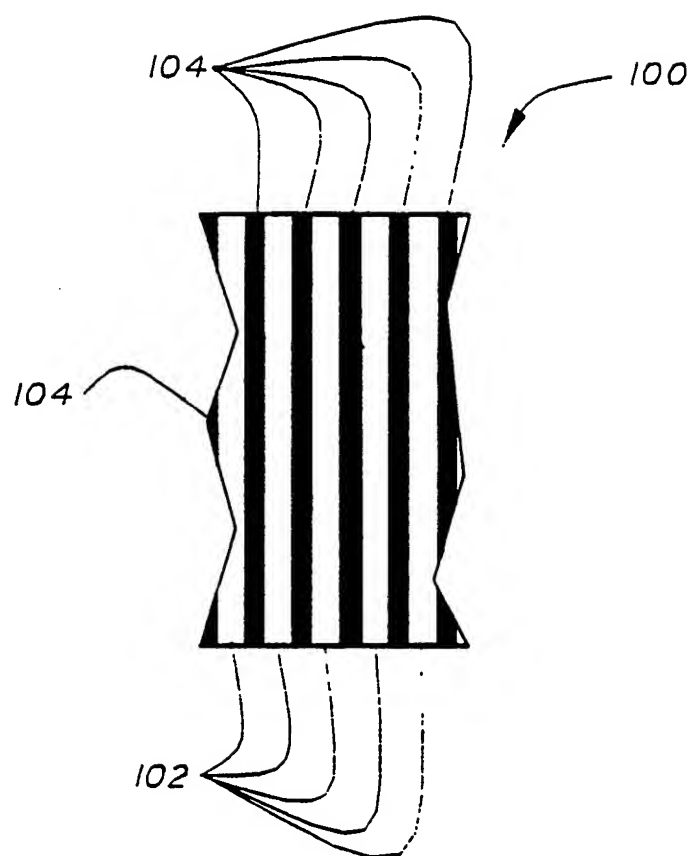


FIG. 2

*FIG. 3*

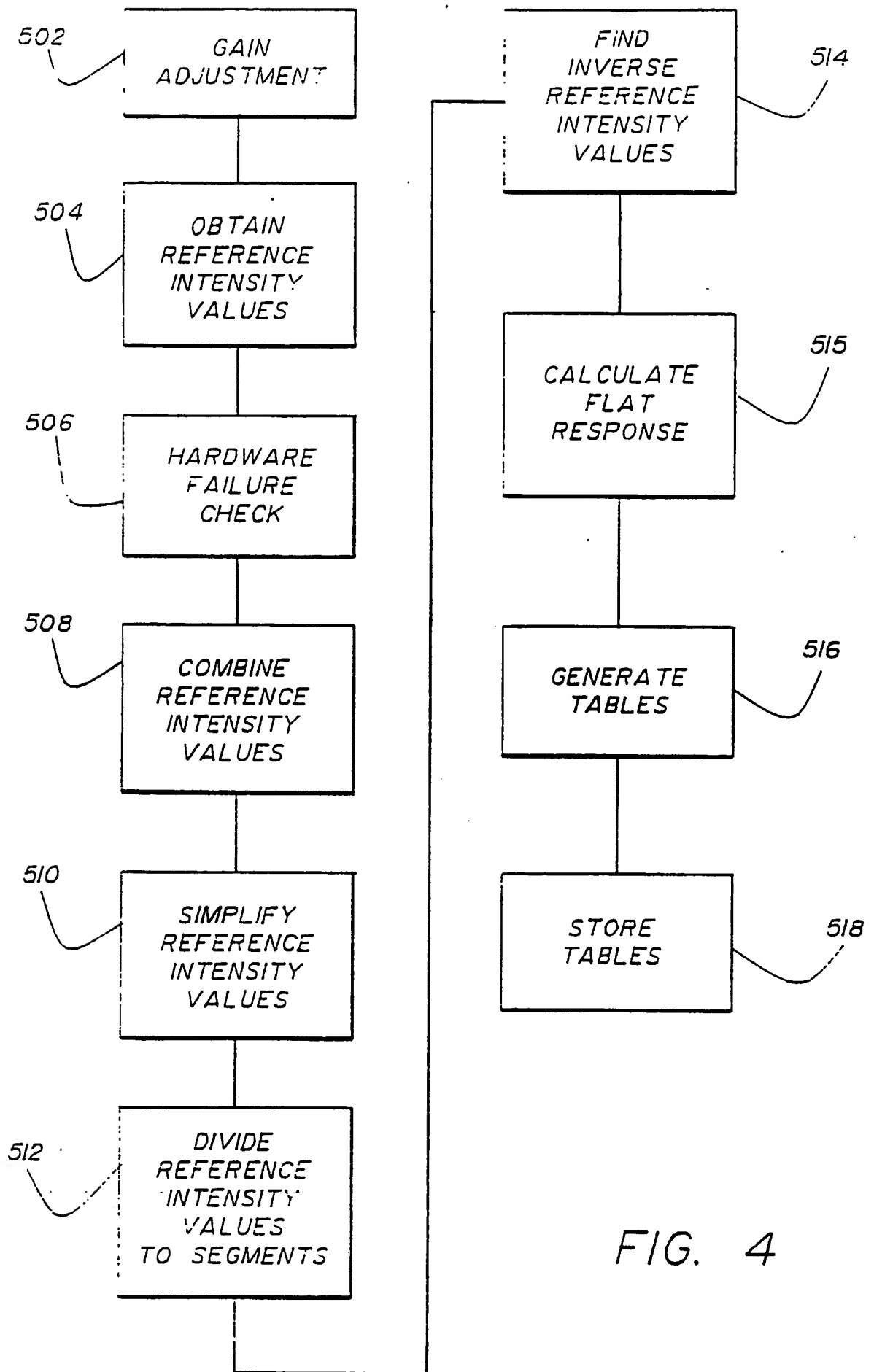


FIG. 4

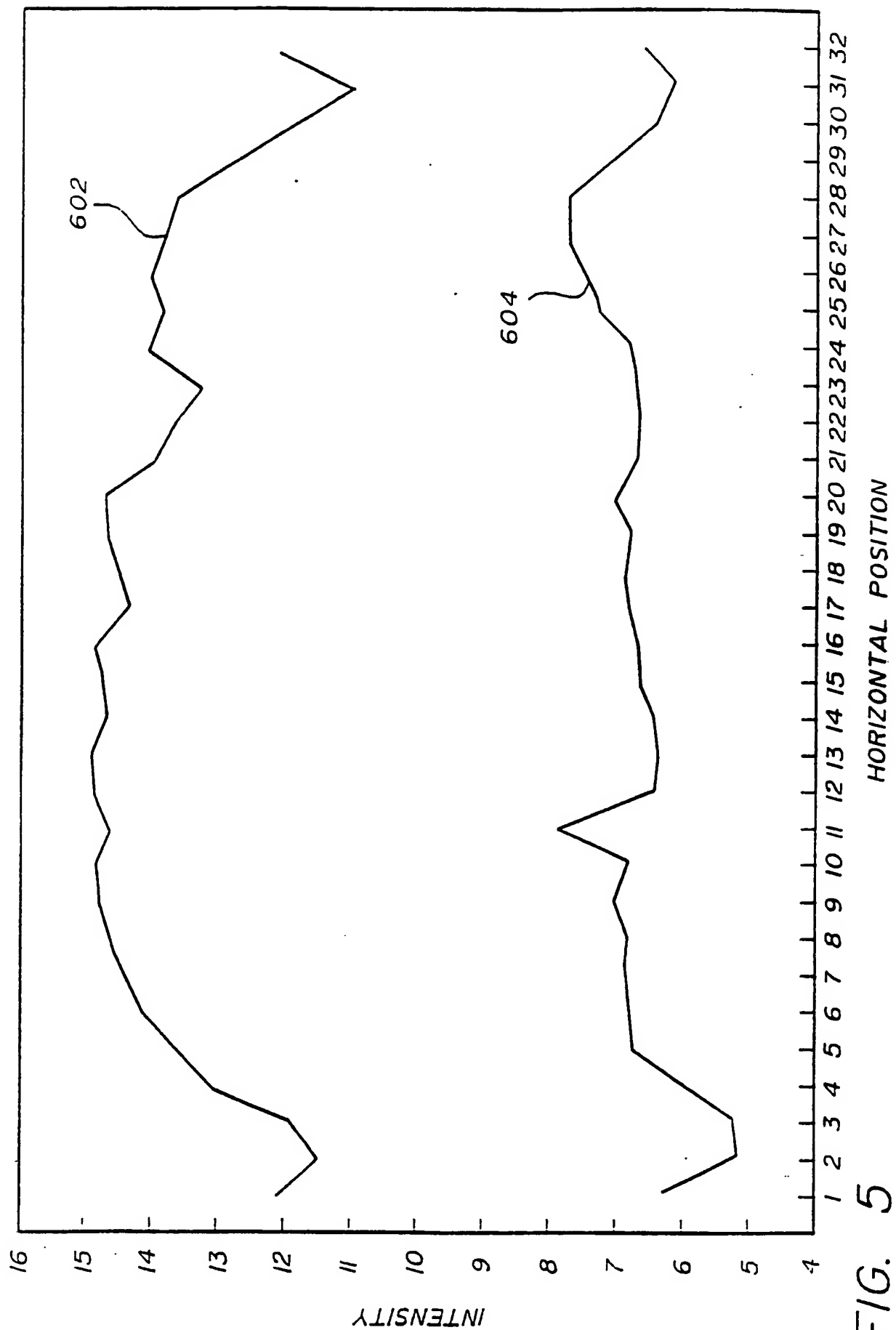


FIG. 5

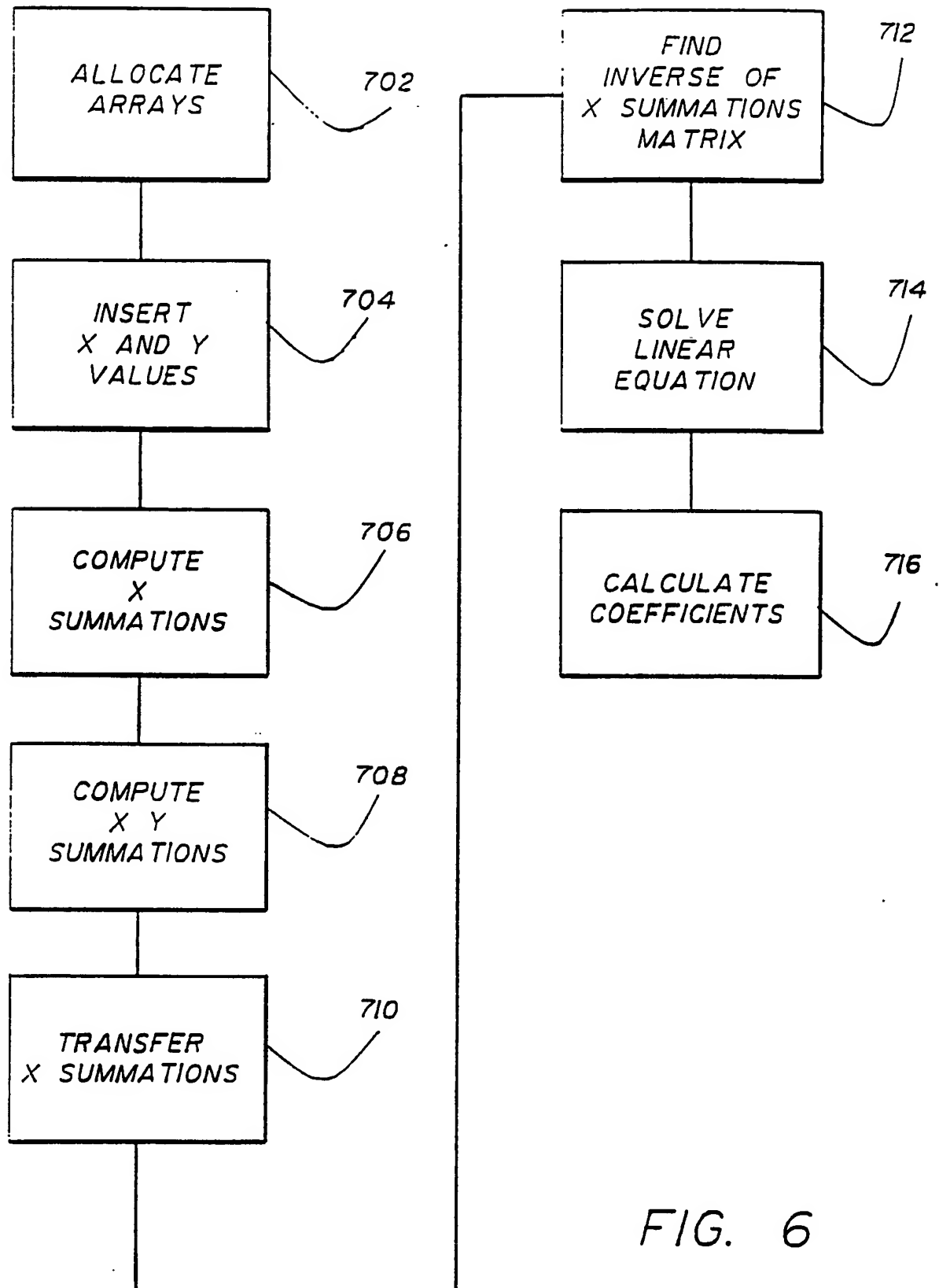


FIG. 6

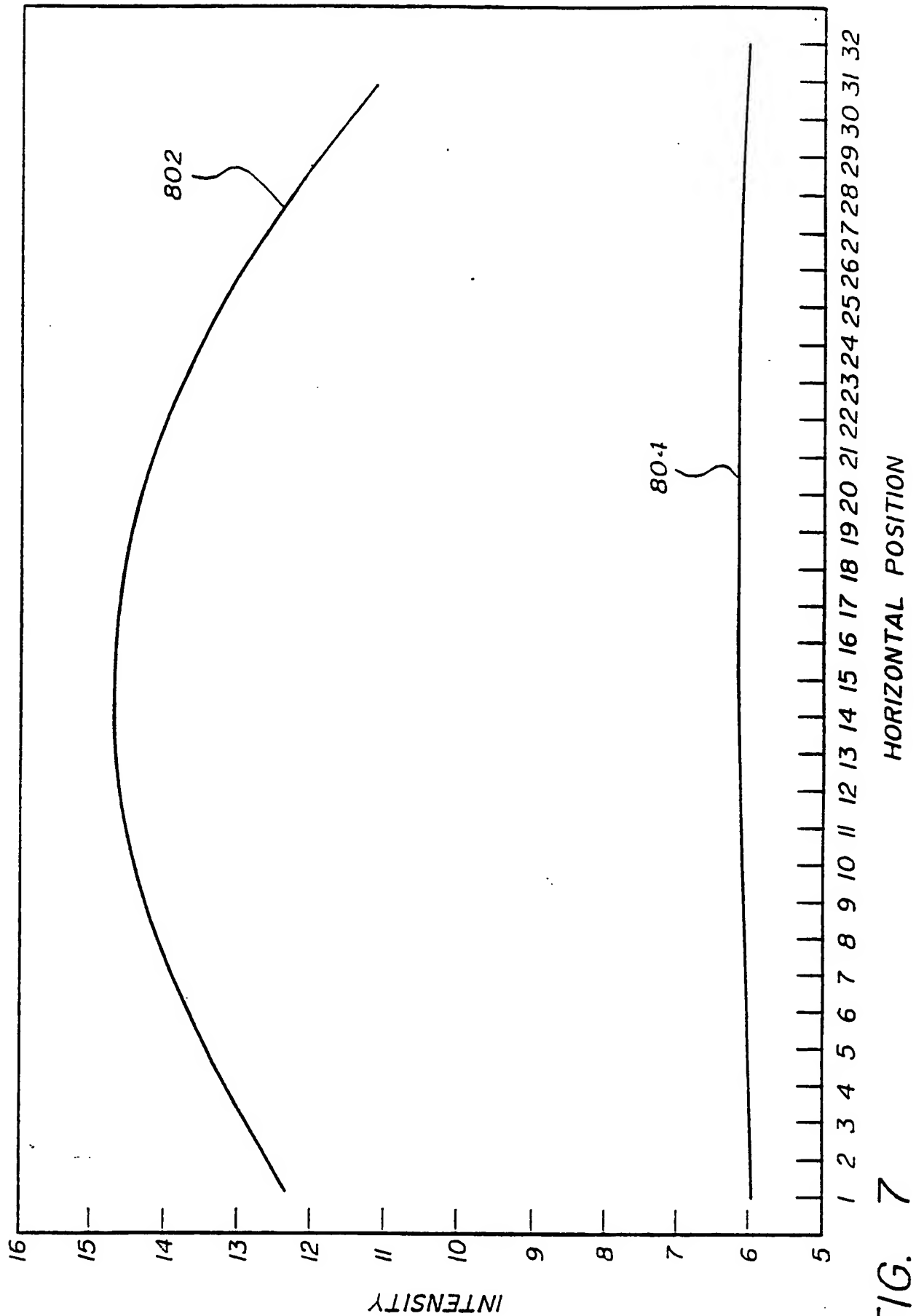
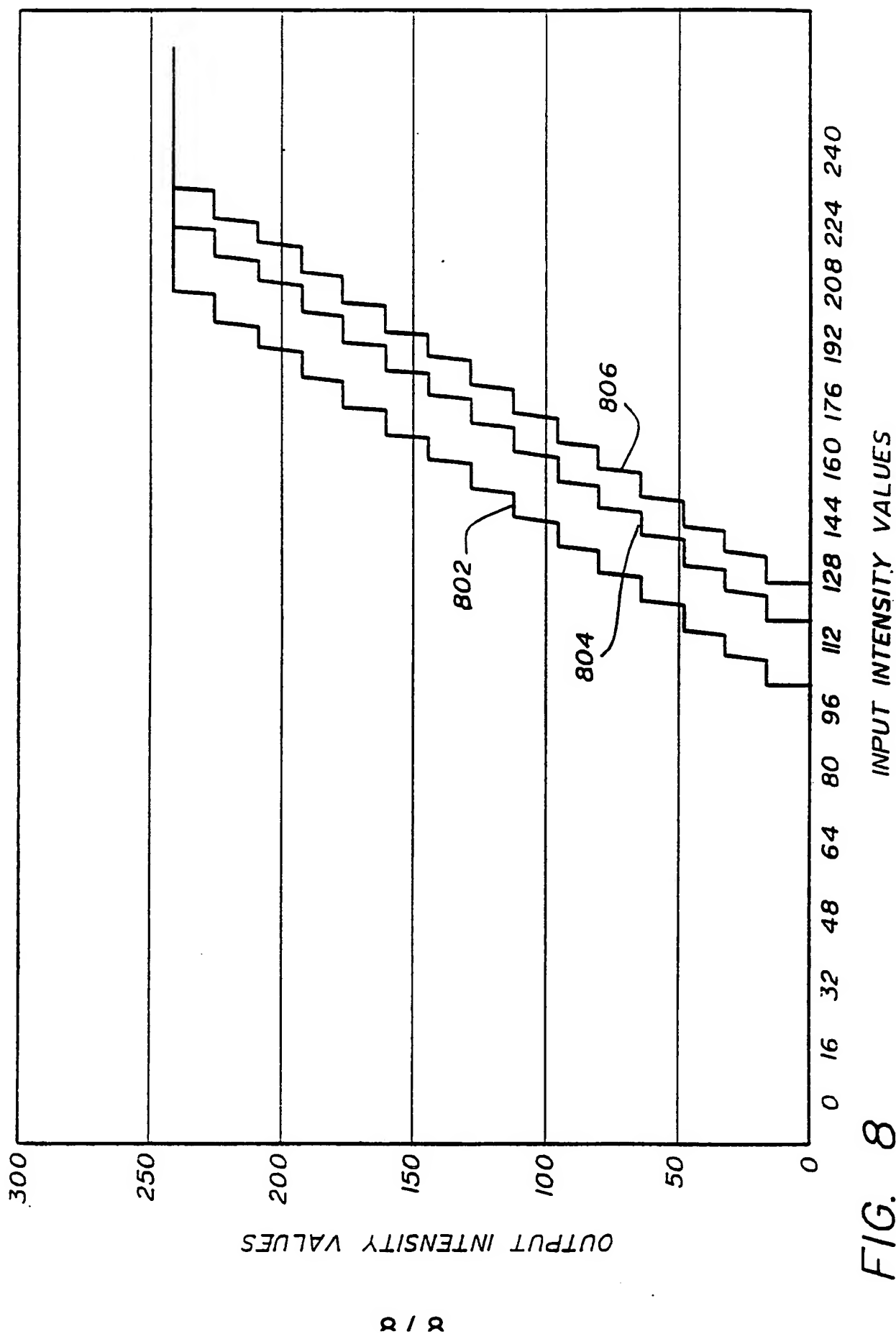


FIG. 7



A. CLASSIFICATION OF SUBJECT MATTER

IPC(5) :G06K 9/38

US CL :382/50

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

U.S. : 382/50, 53

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

none

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US, A, 4,314,281 (WIGGINS) 02 February 1982 (02.02.82). See col. 27, line 12 through col. 28, line 46.	1-9
Y	US, A, 4,430,748 (TUHRO ET AL.) 07 February 1984 (07.02.84). See col. 2, lines 58-68.	1-4
Y	US, A, J. SIMON AND A. ROSENFELD, "DIGITAL IMAGE PROCESSING AND ANALYSIS", Published 1977, Noordhoff International Publishing, pages 122-125.	1-9

☐ Further documents are listed in the continuation of Box C.
 ☐ See patent family annex.

* Special categories of cited documents:	"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"A" document defining the general state of the art which is not considered to be part of particular relevance	"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"E" earlier document published on or after the international filing date	"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)	"Z" document member of the same patent family
"O" document referring to an oral disclosure, use, exhibition or other means	
"P" document published prior to the international filing date but later than the priority date claimed	

Date of the actual completion of the international search

02 NOVEMBER 1992

Date of mailing of the international search report

10 DEC 1992

Name and mailing address of the ISA/US
Commissioner of Patents and Trademarks
Box PCT
Washington, D.C. 20231

Facsimile No. NOT APPLICABLE

Authorized officer

LEO H. BOUDREAU

Telephone No. (703) 305-4762

 RECEIVED
 INTERNATIONAL DIVISION

**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ BLACK BORDERS
- ☐ IMAGE CUT OFF AT TOP, BOTTOM OR SIDES
- ☐ ~~FADED TEXT OR DRAWING~~
- ☒ BLURRED OR ILLEGIBLE TEXT OR DRAWING
- ☐ SKEWED/SLANTED IMAGES
- ☐ COLOR OR BLACK AND WHITE PHOTOGRAPHS
- ☐ GRAY SCALE DOCUMENTS
- ☐ LINES OR MARKS ON ORIGINAL DOCUMENT
- ☐ REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY
- ☐ OTHER: _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.